



CE Linux Forum

Linux DVB API Version 4

<http://www.linuxtv.org>

Michael Hunold
hunold@linuxtv.org



26.01.2005

TOSHIBA

1



Contents

- DVB and Digital TV
- History
- About LinuxTV.org
- Linux DVB API v3 status and problems
- Linux DVB API v4 design and structure
- Concepts
- Current status and todo
- Example
- Helping the project

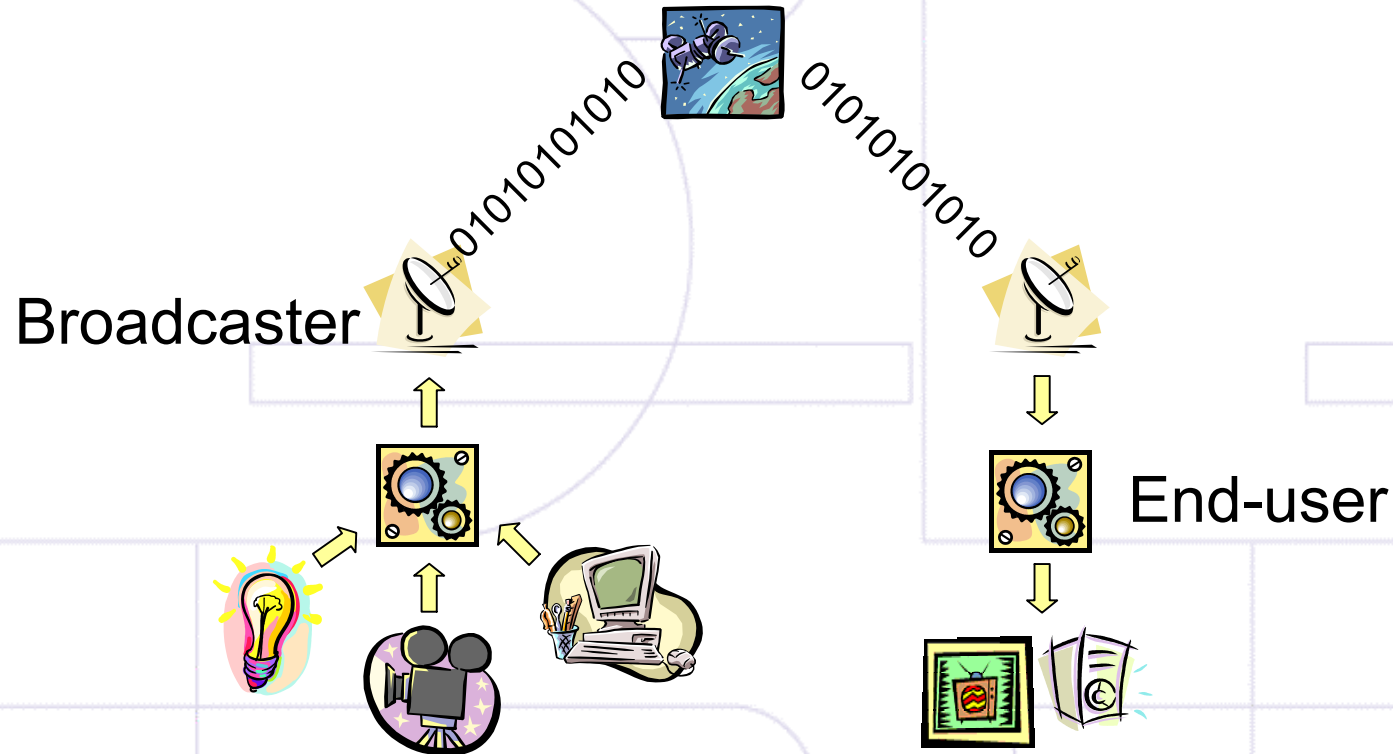


What is DVB?

- Digital Video Broadcast <http://www.dvb.org/>
- delivery of any digitized informations to the home
 - video, audio, subtitles
 - service informations, program guide
 - multimedia applications, ...
- managed by Digital Video Broadcasting Project
 - industry-led consortium of broadcasters, manufacturers, network operators, software developers, regulatory bodies and others
- based on MPEG2 transport streams
- different means of transportation
 - DVB-C ⇨ cable
 - DVB-S ⇨ satellite
 - DVB-T ⇨ terrestrial



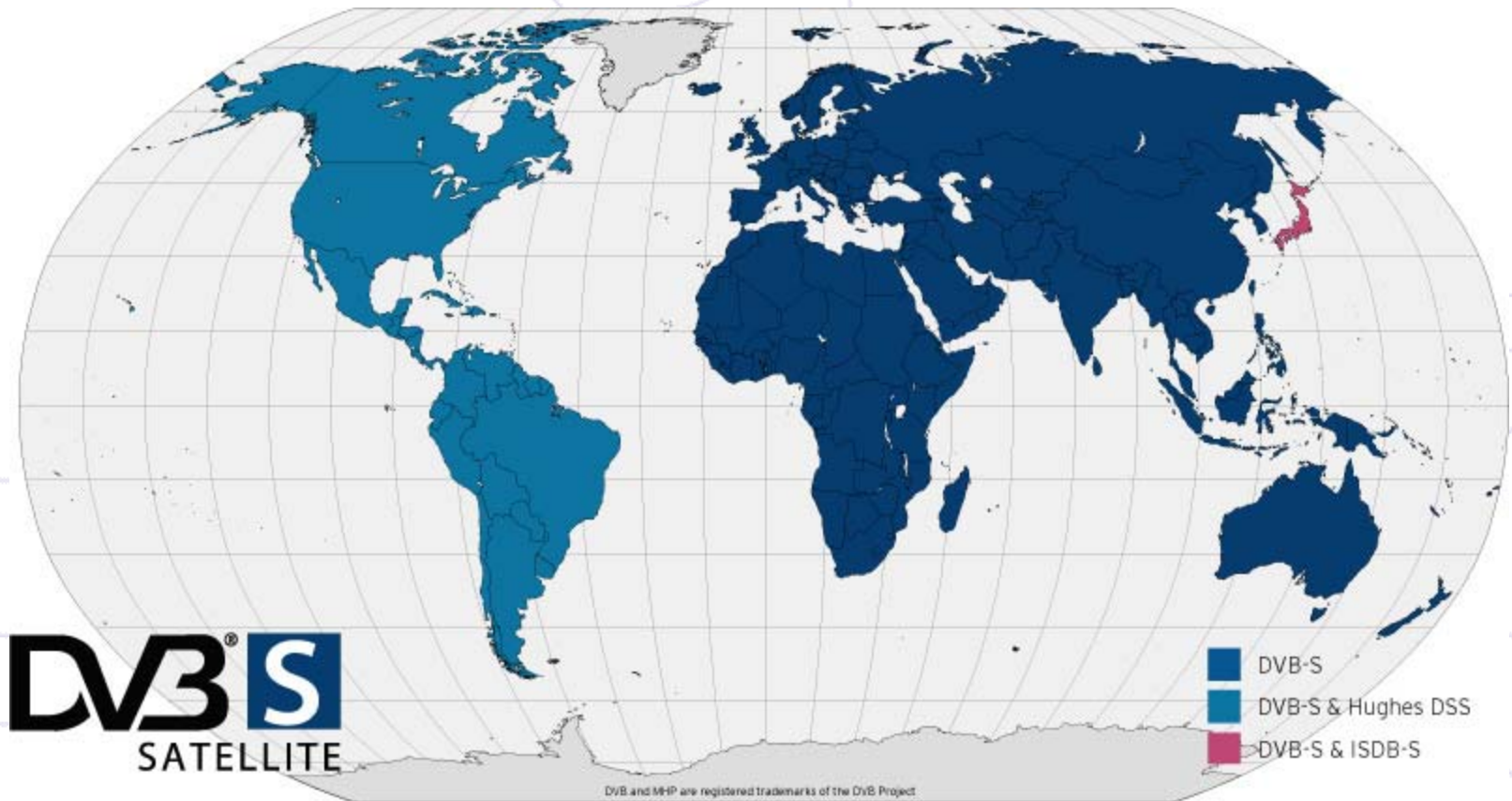
Data flow



- MPEG2 transport stream (TS)
- data packets with 188 bytes



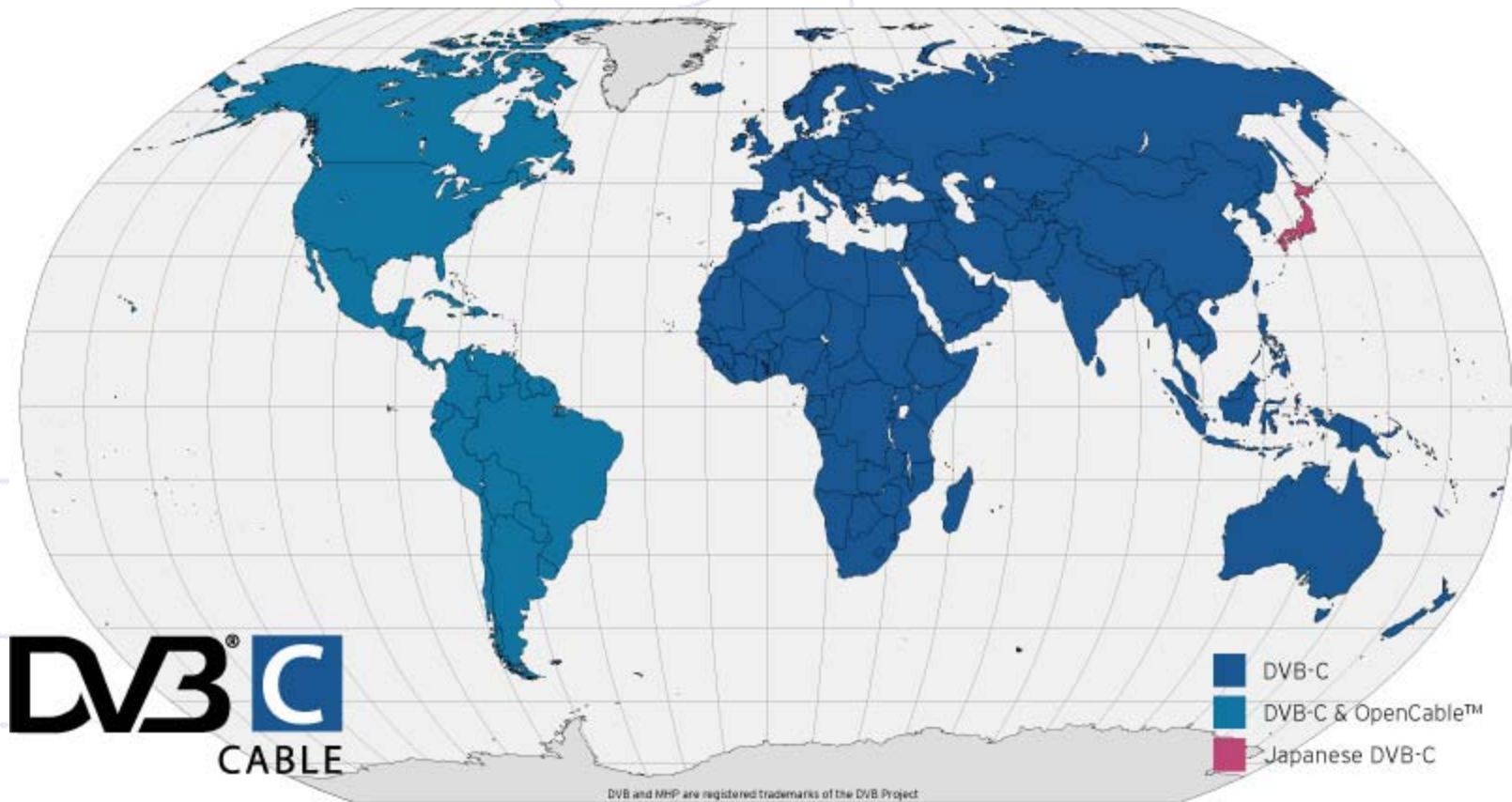
DVB-S adoption map



Used with permission of the DVB project, <http://www.dvb.org>



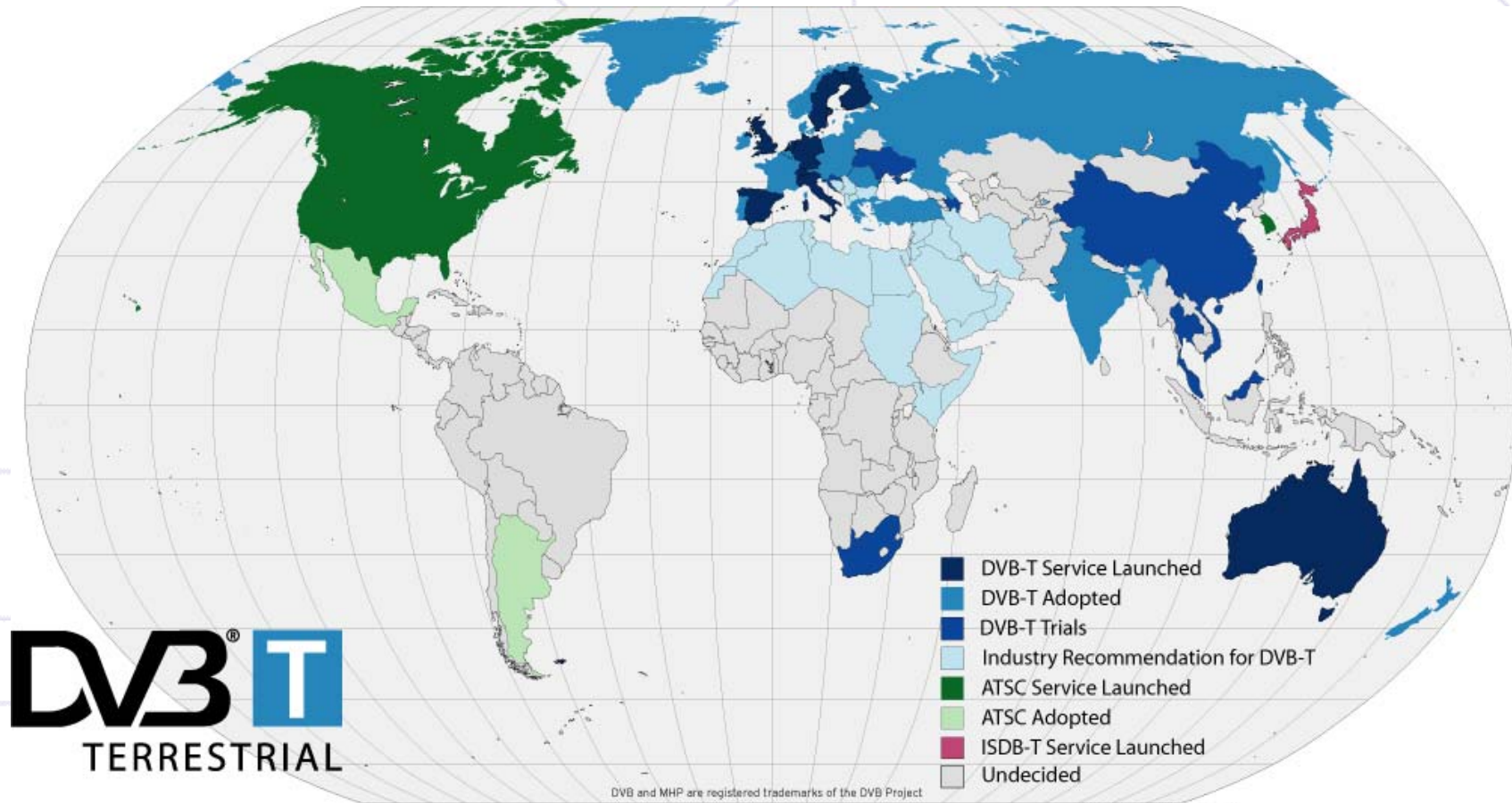
DVB-C adoption map



Used with permission of the DVB project, <http://www.dvb.org>



DVB-T adoption map



Used with permission of the DVB project, <http://www.dvb.org>



Related technologies

- IP-over-DVB (“SkyDSL”, Internet via satellite)
 - MPE (Multi Protocol Encapsulation, ETSI standard)
 - ULE (Ultra Light Encapsulation, IETF draft)
- DVB-over-IP
 - MPEG2 TS distributed via Ethernet
 - RTP streaming
 - DVB-IPI (ETSI TR 102 033)
- general datacasting
 - SSU (System Software Update)
 - proprietary protocols
- DVD playback (MPEG2-PS and MPEG1)



History

- 1998
 - Technotrend GmbH develops popular DVB PCI card
- 1999
 - Siemens produces cards based on Technotrend GmbH design
 - first Linux driver as diploma thesis, ad hoc API based on Video4Linux
- 2000
 - Nokia develops DVB API for <http://www.ostdev.net/>
 - Convergence GmbH implements Nokia DVB API for Siemens card
 - community project LinuxTV.org is launched
- 2001
 - ongoing development results in Linux DVB API v3



History

- 2002
 - Linux DVB API v3 is included in Linux kernel 2.5.44
- 2003
 - Convergence and Toshiba start development of Linux DVB v4 API
 - public discussion of API features on the linux-dvb mailing list
- 2004
 - Linux DVB v4 API is nearly fully specified
 - generic core modules and sample driver for Siemens card available
- 2005
 - first real-world implementation on Toshiba TC90400 MPEG2 decoder chipset („Donau“)



What is LinuxTV.org?

- started and hosted by Convergence GmbH until mid 2004
- community project by DVB enthusiasts and developers interested in Digital TV
- open, independent, non-profit
- hosts other projects related to Digital TV
 - MPEG2 TS/PS multiplexer
 - DSM-CC / MHP object carousel generator



LinuxTV.org goals

- specify complete API for
 - set-top-box (STB)
 - Integrated Digital TV (IDTV)
- standardize support for peripheral hardware
 - as little as possible proprietary APIs and private extensions
- promote existing APIs where possible
 - Linux Input Layer for ir and frontpanel
 - DirectFB for scaler, video encoder, video layer mixer
- create new APIs where necessary
 - analog tv integration
 - avmux
 - smartcard interface
 - video encoder and vbi
 - DVD and subtitle processing unit (SPU) support



Linux DVB

- no complete multimedia framework
 - graphics output and video scalers are handled by DirectFB
 - analog uncompressed video is currently not supported
 - no handling of arbitrary multimedia data
- no support for auxiliary hardware in core API
 - avmux
 - smartcard interface
- means to control digital tv hardware easily and efficiently
 - support for PCI/USB DVB extension cards
 - support for dedicated set-top-box (STB) chipsets
 - support for integrated digital TV (IDTV) solutions
- hardware independent driver framework
- kernel level only



Linux DVB

- Motivation
 - make life of software and hardware developers easier
 - provide a consistent abstraction layer for different hardware
- Benefits
 - Software companies
 - easier support for different hardware platforms
 - hardware independent applications and middleware
 - Hardware vendors
 - easier software support for existing chipsets
 - smooth transition from one chipset generation to the next
 - new business opportunities by faster time-to-market



Linux DVB API v3 users

- Open-source projects
 - PC
 - vdr (video disk recorder)
 - mplayer and xine
 - kaxtv, mythtv, klear, xawtv 4.0
 - Embedded Linux platforms on PowerPC-based „dbox2“
 - Neutrino, Enigma ⇒ <http://wiki.tuxbox.org/> (german)
 - platform independent
 - dvbstream, dvbsnoop
- Commercial users
 - Dreambox using IBM Pallas/Vulcan (PowerPC)



Present situation

- PCs and embedded platforms are diverging
 - “budget” PCI cards on PCs
 - provide full, raw TS
 - all decoding and processing is done by main CPU
 - Embedded platform STB/IDTV chipsets
 - data is demultiplexed for direct application use
 - specialized hardware or firmware relieves main CPU
 - no new „full-featured“ DVB PCI/USB card is in sight
 - new challenges with supporting embedded platforms running Linux
- ⇒ Linux DVB v4 API heads towards highly-integrated embedded STB and IDTV systems



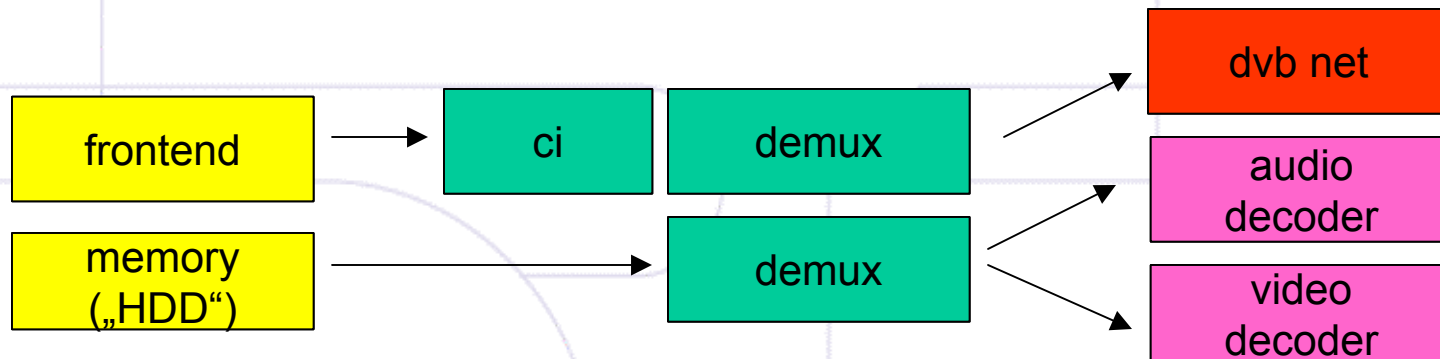
Linux DVB API v3 problems

- focussed on Siemens DVB card
- namespace inconsistencies
- inconsistent remains of ad-hoc DVD/SPU support
- legacy OSD API designed for Siemens DVB card only
- superfluous internal DVB kernel demux layer
- very limited support for modern hardware
 - multiple frontends, video and audio decoders
- explicit source-sink connections not possible
- no support for special recording hardware and event logging
- all data transfers through ringbuffers, no zero-copy DMA



Linux DVB API v4 design

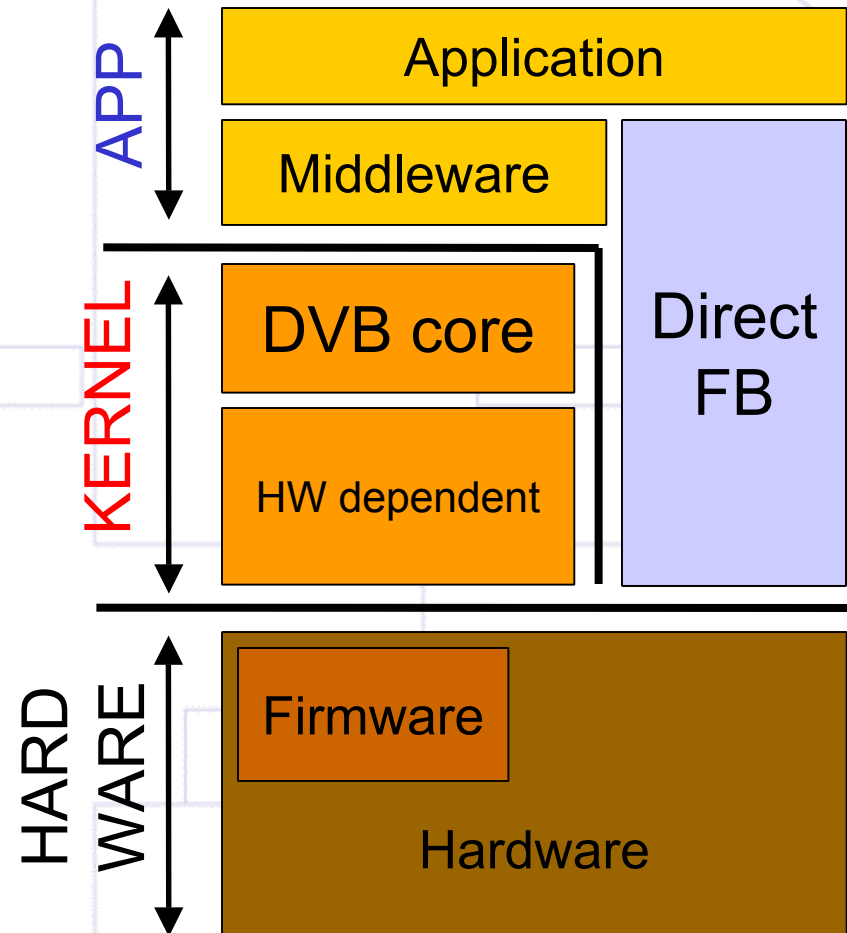
- Linux/Posix character device interface `/dev/dvb/adapter0/...`
 - input: frontend, memory
 - processing: ci, demux
 - decoding: video, audio
 - output: audio mixer, spdif, dvb-net
- source/sink connection via I/O controls
- zero copy DMA via `mmap()`





Implementation structure

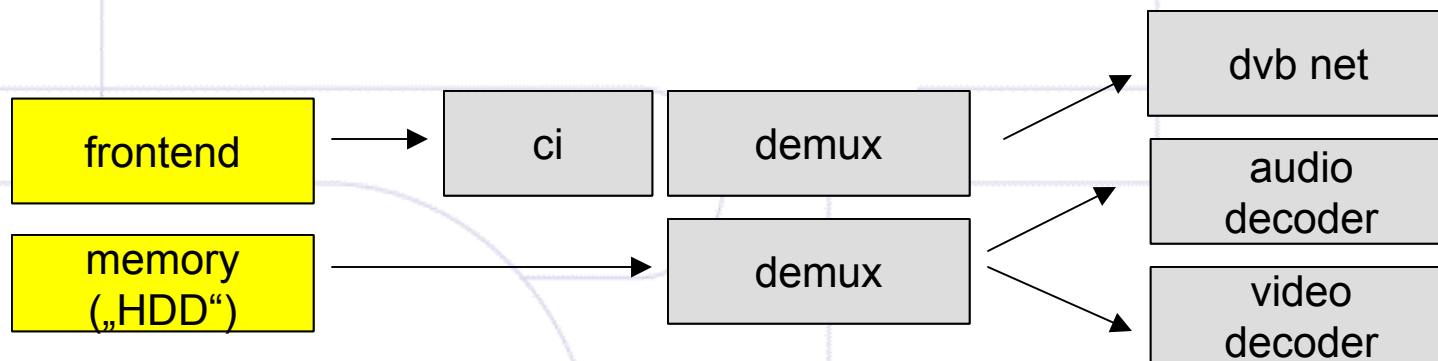
- DVB core
 - implements hardware independent functionality
 - does sanity checks and parameter checking
 - enforces policy restrictions
 - different levels of abstraction
 - demux, frontend (high)
 - video (low)
- external APIs
 - graphics, scaler and video output handled by DirectFB





Input devices

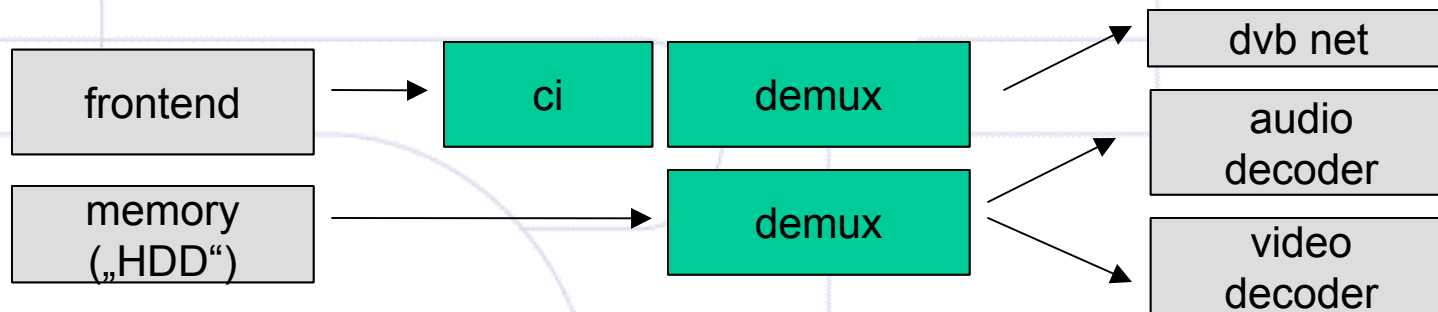
- provide data for stream processing
- frontend input
 - controls tuner and demodulator hardware
 - supports satellite (DVB-S), cable (DVB-C) and terrestrial (DVB-T)
 - in case of satellite equipment (LNB, dish rotor) control via Eutelsat's DiSEqC protocol
- memory input
 - zero-copy DMA for HDD playback or network streaming





Processing devices

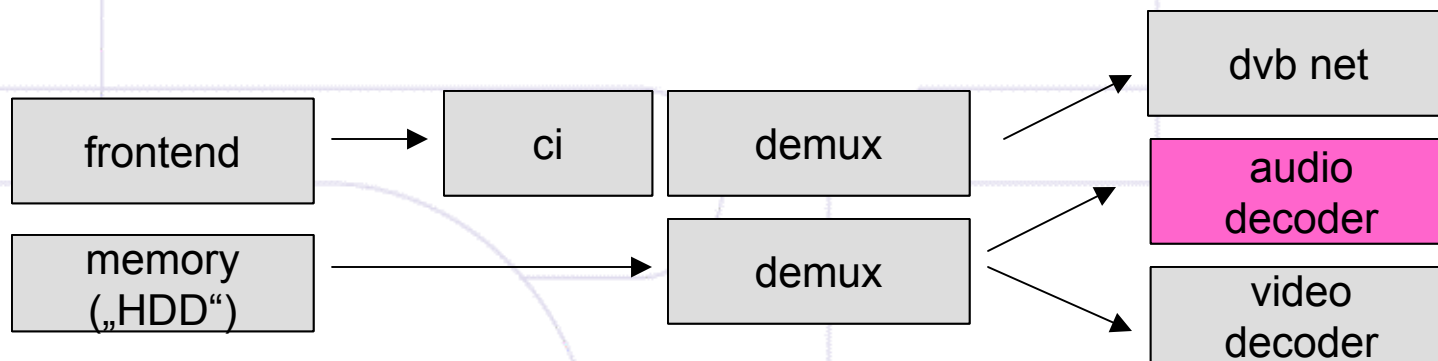
- ci (common interface)
 - slot handling for common interface (CI) PCMCIA slot
 - simple transport layer message interface
- demux (demultiplexer)
 - processes one input stream from frontend or memory input
 - outputs to multiple destinations
 - PID filter (TS packet or payload only), DVB section filter
 - PS and multiplexed PES input with stream id filter
 - recording filter for HDD recording with event logging support
 - decoding feeds for live tv (video, audio, teletext/VBI insertion)





Decoding devices

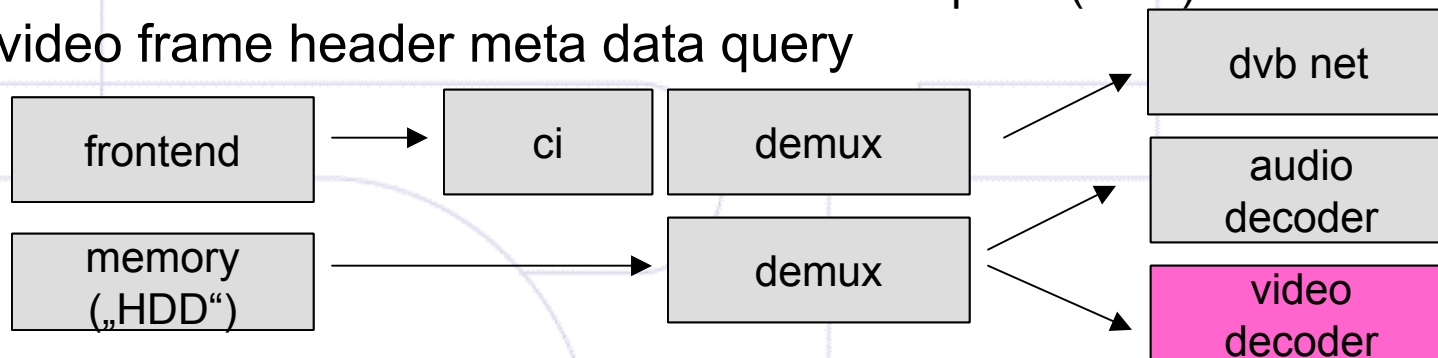
- audio
 - ALSA and OSS are not suitable for embedded STBs/IDTVs
 - provides decoding and downmixing
 - provides PCM playback and optional MP3 playback
 - optional postprocessing
 - virtual surround
 - dynamic compression etc.
 - S/P-DIF and I2S inputs





Decoding devices

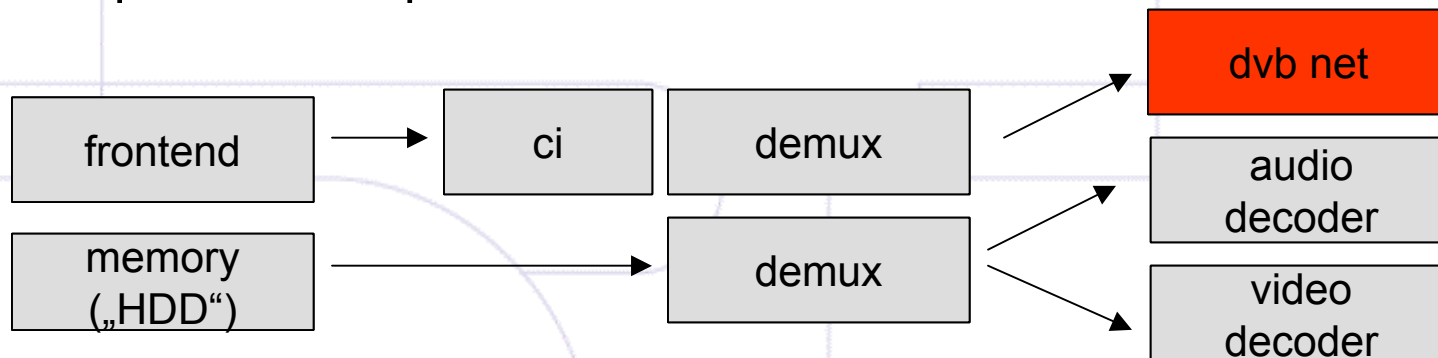
- video
 - usually MPEG 1/2 decoder
 - limited presentation support
 - auto-scaling, pan & scan
 - trickmode support for memory input
 - fast forward, (fast) reverse, slowmotion
 - stillpicture and dripfeed support
 - video event handling
 - sequence header changes
 - extract user data like active format description (AFD)
 - video frame header meta data query





Output devices

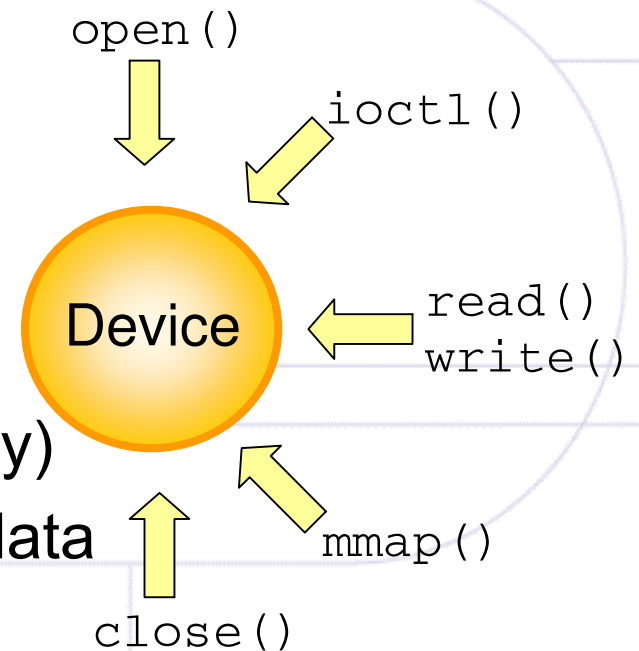
- (audio) mixer
 - mix 5.1 and stereo signals to TV, VCR and headphone outputs
 - tone control, speech enhancement, test tone generator, ...
- spdif
 - S/P-DIF header control
 - connectable to various sources
- net
 - MPE and ULE decapsulation
 - data packets are provided to the Linux network stack





Control concept

- POSIX device interface
- `open()`/`close()` to access devices
 - with “write access” once
 - with “read only” multiple times
- `ioctl()` to control device
- `read()` to pull stream (section filter only)
- `mmap()` for zero-copy DMA access to data
 - get recording data
 - provide playback data



```
fd = open('/dev/dvb/adapter0/video0', O_WRONLY);  
ioctl(fd, DVB_VIDEO_PLAY, 1000);
```



Connection concept

- DVB_XXX_SET_SOURCE I/O control connects devices to input sources by passing a file descriptor
- common for all processing, decoding and output devices
- only possible on a device open with write permissions



```
src = open(`/dev/dvb/.../frontend0', O_RDONLY);  
dst = open(`/dev/dvb/.../demux0', O_WRONLY);  
ioctl (dst, DVB_DEMUX_SET_SOURCE, &src);
```



Capability concept

- different hardware has different capabilities
- sometimes even different devices of the same kind have different capabilities
- applications need to be able to distinguish devices and hardware
- DVB_XXX_GET_CAPS IOCTL exposes device capabilities



```
vcaps.cap = DVB_VIDEO_CAP_SOURCE_FORMATS;  
ioctl(fd, DVB_VIDEO_GET_CAPS, &vcaps);  
if (vcaps.val & DVB_VIDEO_XXX_CAPABLE) {  
    // device supports xxx
```



Filter concept for demux devices

- opening the demux `O_WRONLY` offers the demux device itself
- opening the demux `O_RDONLY` adds a new filter
 - section filter
 - recording filter
 - PES filter
 - decoding feeds
- specific I/O controls set filter parameters or change them
- closing the filedescriptor removes filter



```
struct dvb_demux_section_filter f;  
flt = open(`/dev/dvb/adapter/demux0', O_RDONLY);  
[... set filter options...]  
ioctl(flt, DVB_DEMUX_SET_SECTION_FILTER, &f);  
[... do something ...]  
close(flt);
```




Current Linux DVB API v4 status

- available via CVS from linuxtv.org (“dvb-kernel-v4”)
- supports 2.4 and 2.6 kernels
- major requested functionality is implemented
- continuing discussion on Linux DVB mailing list
 - current and future API development
 - remaining implementation details
- prototype driver for Siemens card
 - PC platform for easy evaluation
 - software emulation to compensate for missing hardware support
- API specification document is work-in-progress
 - auto generated version with few annotations is available
- complete software emulation of DVB device is planned
 - overcome Siemens card drawbacks (no multiple tuner support)



Linux DVB todo

- Linux 2.6 kernel integration
- finish API specification
 - multi client support for section filter data
 - CI routing
 - support for other MPEG2 TS based formats
 - ATSC (USA)
 - ISDB (Japan)
- CE Linux Forum
 - discuss Linux DVB in technical working group



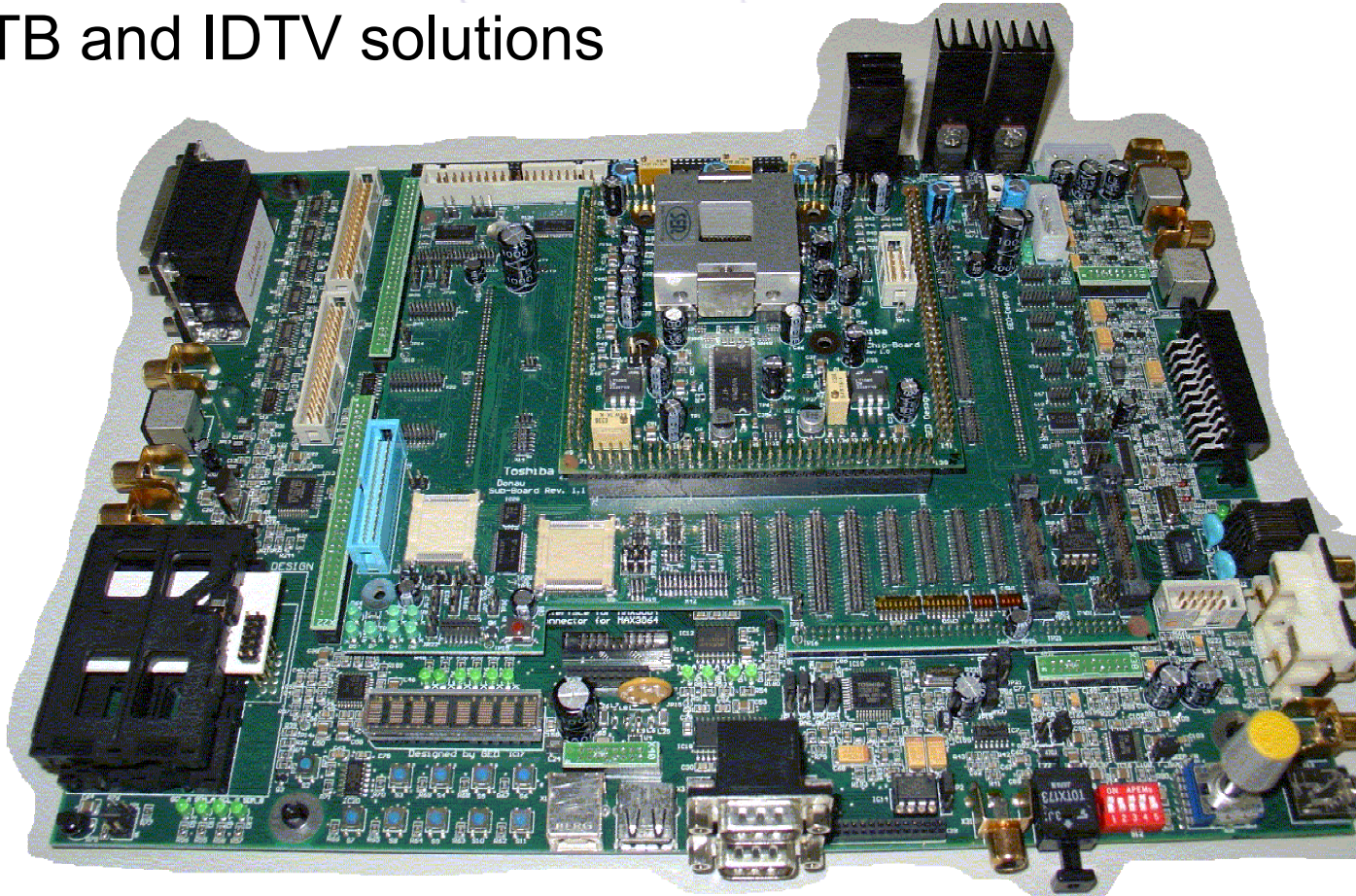
Helping Linux DVB and LinuxTV.org

- open, vendor independent, non-profit project
 - discussions and proposals are very welcome
 - thrives on the contribution of its participants
 - follows the spirit of Linux development model
 - always open to receive proposals for API enhancements
- What you can do
 - subscribe to the linux-dvb mailing list
 - engage in discussions and share your opinion
 - do code review and send patches to the mailing list
 - get a CVS account for larger contributions
 - don't wait for others to do your work



Real world example

- Toshiba TC90400 „Donau“ Digital TV System-on-a-Chip for STB and IDTV solutions





Thank you

- Thank you for your attention
- Any questions?

Related Links

- <http://www.linuxtv.org>
- <http://www.directfb.org>

26.01.2005



Abbreviations

- API = application programming interface
- CI/CA= common interface, common access
- CVS = concurrent versioning system
- DMA = direct memory access
- DSM-CC = digital storage media command and control
- DVB = digital video broadcast
- HDD = hard disk drive
- IDTV = integrated digital television
- MHP = multimedia home platform
- OSD = on-screen display
- PES = packetized elementary stream
- PS = program stream
- SPU = subtitle processing unit
- S/P-DIF = Sony/Philips digital interface
- STB = set top box
- TS = transport stream

26.01.2005